AMENDMENTS TO THE DRAWINGS

Amended Formal Drawings are enclosed, with each sheet identified as a "Replacement Sheet".

REMARKS

Claims 1-19 are pending in the current application. In an office action dated November 7, 2007 ("Office Action"), the Examiner objected to the drawings, objected to the specification, rejected claim 19 under 35 U.S.C. § 101, and rejected claims 1-19 under 35 U.S.C. § 102(b) as being anticipated by Karger et al., U.S. Patent No. 4,787,031 ("Karger"). Applicants' representative has provided a corrected drawing to address the Examiner's objections to the drawings, and has amended the specification, above, to properly refer to the corrected drawing. With regard to the Examiner's objection to the specification related to acronyms, Applicants' representative first points out that the Examiner does not provide any statute, case law, or rule to support the objection, and, therefore, the objection has been made without authority and is unjustified. More importantly, Applicants' representative wishes to respectfully point out that the term "*vmsw*" is not used as an acronym in the specification. The term "*vmsw*" is simply the name of a proposed Intel Itanium® instruction, just as *mov*, *pop*, *sub*, *bsf*, and *cld* are names of instructions in the Intel 80X86 processor family. The term "*vmsw*" first occurs in the summary-of-the-invention section of the current application, on line 22 of page 6. In the sentence including the first occurrence of the term "*vmsw*," the *vmsw* instruction is quite clearly and completely defined. In Applicants' representative's respectfully offered opinion, the Examiner's objection to the term "*vmsw*" is both unjustified and unjustifiable. With regard to the Examiner's 35 U.S.C. § 101 rejection of claim 19, Applicants' representative has cancelled claim 19, in the above amendment. Applicants' representative respectfully traverses the rejections of the remaining claims, 1-18, as being anticipated by Karger under 35 U.S.C. § 102(b).

First, although not dispositive with regard to the 35 U.S.C. § 102(b) rejections, Applicants' representative points out that, in the first two sentences of the detailed-description-of-the-invention section, beginning on line 16 of page 8 of the current application, Applicants have quite explicitly indicated that the current invention is directed to architectural extensions to the Intel Itanium® processor family and other, comparable modern computer processors with architectural features similar to those of

the Intel Itanium processors. By contrast, Karger is directed to features of an old processor architecture, namely that of the VAX-11 processor architecture produced in the late 1970's and 1980's by the Digital Equipment Corporation. While not dispositive with regard to the rejections, this initial portion of the detailed description of the invention would certainly lead one to expect that the current invention is probably not directed to, or related to, older processor architectures, such as the VAX architecture.

Claim 1 claims "a virtualization-mode-switch instruction that switches the state of the processor between virtualization-mode and non-virtualization-mode without incurring an interruption." Support for this claim element can be found in numerous places in the current application, but it is probably most concisely provided beginning on line 17 of page 17 of the current application. The intent of the new *vmsw* instruction, in one embodiment of the current invention, is to allow a guest operating system to directly enter virtual-machine-monitor mode without incurring an interruption. As the Examiner hopefully appreciates, interruptions interrupt sequential instruction execution by a processor and redirect execution to a short routine or code sequence known as a "handler." Interruptions are carried out by processor-hardware-level features. The term "interruption" refers to a general concept that includes various types of interruptions, including interrupts, faults, traps, and exceptions. In general, the interruption mechanism provided by processor architectures re-adjusts stack pointers, privilege level, and other values stored in routine-context-associated registers and status registers in preparation for transfer of execution to an interrupt handler, fault handler, exception handler, or trap handler, generally including raising the privilege level to a more privileged privilege level than the privilege level of the currently executing routine.

The Examiner cites lines 62-65 of column 20 of Karger as teaching the virtualization-mode-switch instruction of claim 1. These lines of column 20 refer to Figures 8A-1 and 8A-2 of Karger, which depict, in detail, a change-operating-mode instruction provided by a VAX architecture proposed by Karger and described in the cited lines of Karger. Were the Examiner to carefully review these figures, the Examiner would find that the change-operating-mode instruction proposed by Karger has only three possible results: (1) return of an error, as shown in the second step of Figure 8A-1, when

the processor-status-longword field *IS* is not set when the processor is not in virtual-monitor mode, as determined in the first step shown in Figure 8A-1; (2) raising of a change-mode exception, the final step of Figure 8A-1; and (3) raising of a VM-emulation exception, as shown in the final step of Figure 8A-2. An error either constitutes a system crash or causes a different interruption. The change-mode and VM-emulation exceptions are interruptions. Thus, Karger not only fails to teach the virtualization-mode-switch instruction of claim 1, Karger specifically teaches away from a virtual-mode-switch instruction "that switches a state of the processor between virtualization-mode and non-virtualization-mode ***without incurring an interruption***." Because claims 2-3 depend from claim 1, claims 2-3 cannot possibly be anticipated by Karger.

Claim 4 includes the element "a virtualization fault invoked by the computer processor when a routine executing in virtualization mode at a highest privilege level attempts to execute an instruction needing virtualization." For teaching this element, the Examiner refers to lines 3-10 of column 5 of Karger. Lines 3-10 of column 5 of Karger state:

> First, the processor determines that the instruction is executable in the current operating mode. Second, the processor determines that the operands, if any, are in memory, and that they are in pages that are available to the operating mode in which the program is running; that is, the processor determines that a page fault or access violation will not occur when it attempts to retrieve the operands.

These cited lines have absolutely nothing whatsoever to do with a virtualization fault. In fact, this portion of Karger discusses prior-art VAX-11 architecture features related to protection rings. It most certainly has nothing to do with a special virtualization-mode fault that is invoked by a processor when a highest-privilege-level virtualization-mode routine attempts to execute a privileged instruction. The cited portion of Karger does not teach, mention, or suggest the above-mentioned element of claim 4, and thus fails to anticipate either claim 4 or claims 5-6, which depend from claim 4.

In rejecting claim 7, the Examiner cites lines 46-50 of column 9 of Karger as teaching "a flexible highest-implemented-virtual-address bit that, in virtualization mode, is checked by the processor and reported by the virtual machine monitor to be less than the highest-implemented-virtual-address bit in the non-virtualization mode, so that a

high-order portion of virtual-address space is accessible only to a virtual machine monitor executing in non-virtualization mode." Lines 46-50 of column 9 of Karger state:

> In accordance with the invention, the virtual machine monitor 20 also provides virtual operating modes providing four protection rings, including a virtual user ring, which corresponds to the real user ring, a virtual . . .

This portion of Karger teaches nothing whatsoever to do with virtual-address space, a bit related to virtual-address space, or anything at all to do with memory. This portion of Karger simply indicates that the proposed VAX architecture provides four protection rings, or privilege levels, for routines executing under virtual-operating mode. Claim 7 is clearly not anticipated by Karger.

The method claims 8-18 all include language similar to the language included in the above-discussed computer-processor claims, and therefore cannot possibly be anticipated by Karger for the same reasons that the above-discussed computer-processor claims are not anticipated. Applicants' representative cannot understand why the Examiner has cited portions of Karger that have no possible relation to elements of the current claims and do not even contain similar term and phrases. The current application is directed to processor-architecture features to facilitate efficient virtual-machine-monitor implementation on modern processors. The currently claimed highest-implemented-virtual-address bit has nothing whatsoever to do with privilege levels or protection rings. The currently claimed virtualization-mode-switch instruction is clearly claimed to not generate an interruption, but Karger's change-mode-instruction implementation shown in Figures 8A-1 and 8A-2 of Karger includes classic preparation-for-interruption-related steps in which stack pointers are adjusted to prepare for ***explicitly-shown*** hardware exceptions. A discussion of general tests related to instruction execution by a processor has nothing to do with the currently claimed specialized virtualization fault.

In Applicant's representative's opinion, all of the claims remaining in the current application are clearly allowable. Favorable consideration and a Notice of Allowance are earnestly solicited.

Respectfully submitted,
Jonathan K. Ross et al.
Olympic Patent Works PLLC

Robert W. Bergstrom
Registration No. 39,906

Enclosures:
Postcard
Transmittal in duplicate

Olympic Patent Works PLLC
P.O. Box 4277
Seattle, WA 98194-0277
206.621.1933 telephone
206.621.5302 fax